

**UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA
(UNAN-LEÓN)**

FACULTAD DE CIENCIAS Y TECNOLOGÍAS.

INGENIERIA EN TELEMATICA.



COMPONENTE: SAAS.

Grupo 2 Telemática

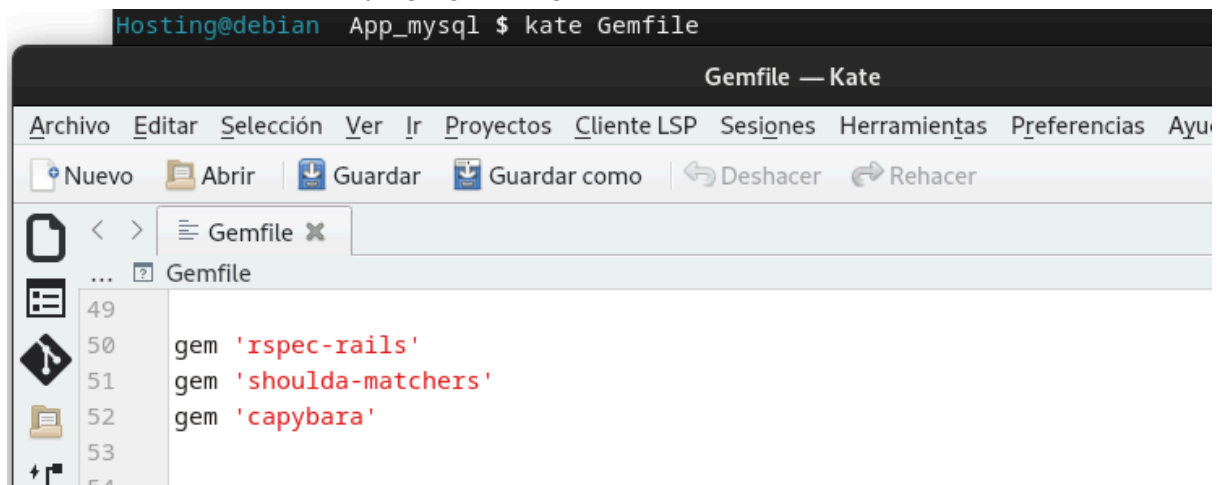
INTEGRANTES:

- Hosting Stewart Lopez Martinez.

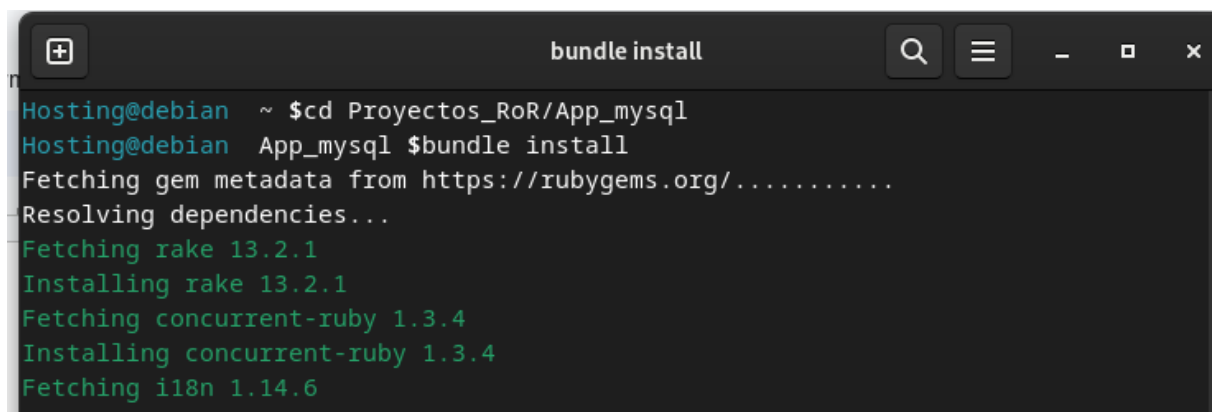
DOCENTE: Ervin Montes

**¡A LA LIBERTAD POR LA UNIVERSIDAD!
“AÑO LECTIVO 2024”**

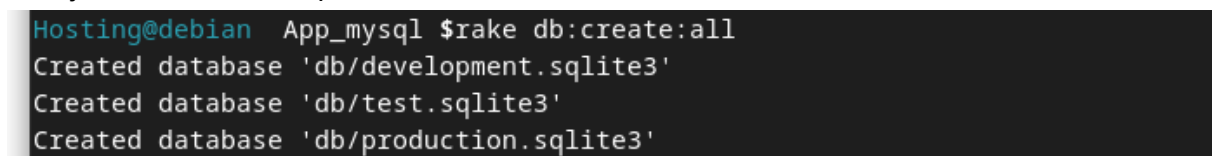
1. Crea un proyecto rails
2. Prepara el escenario de test
 - 2.1 Abrir el archivo Gemfile y agregar lo siguiente



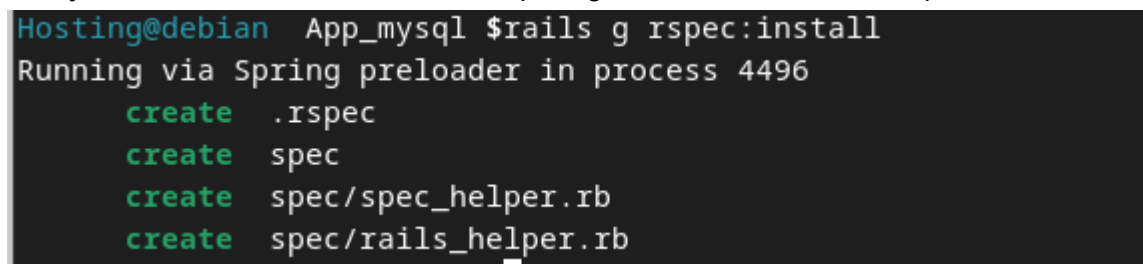
- 2.2 Ejecutar el siguiente comando



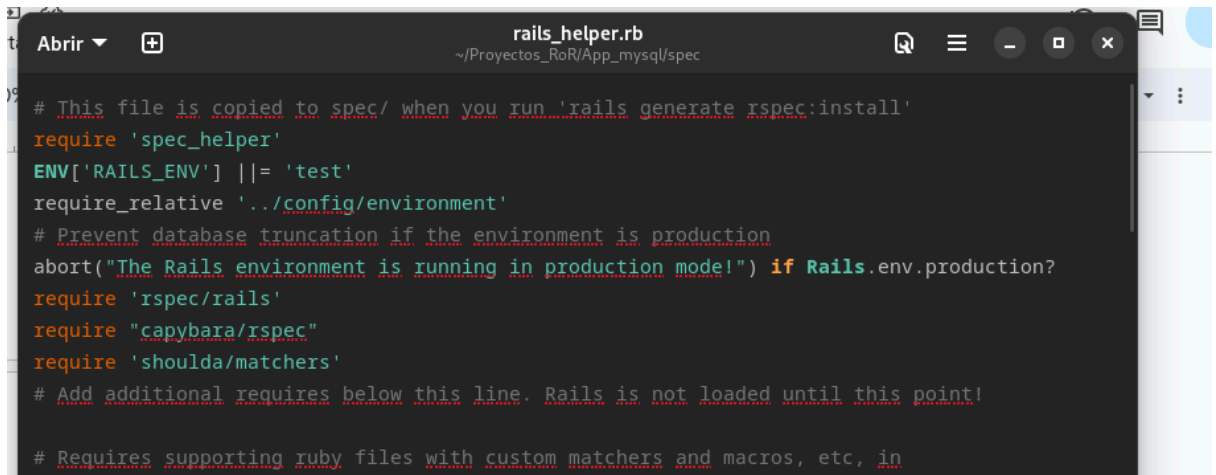
- 2.3 Ejecuta el comando par crear la base de datos



- 2.4 Ejecutar el comando en la terminal, para generar los archivos de rspec.



2.5 Como parte de la configuración hay que agregar las siguientes líneas dentro del archivo spec/rails_helper.rb que fue generado anteriormente

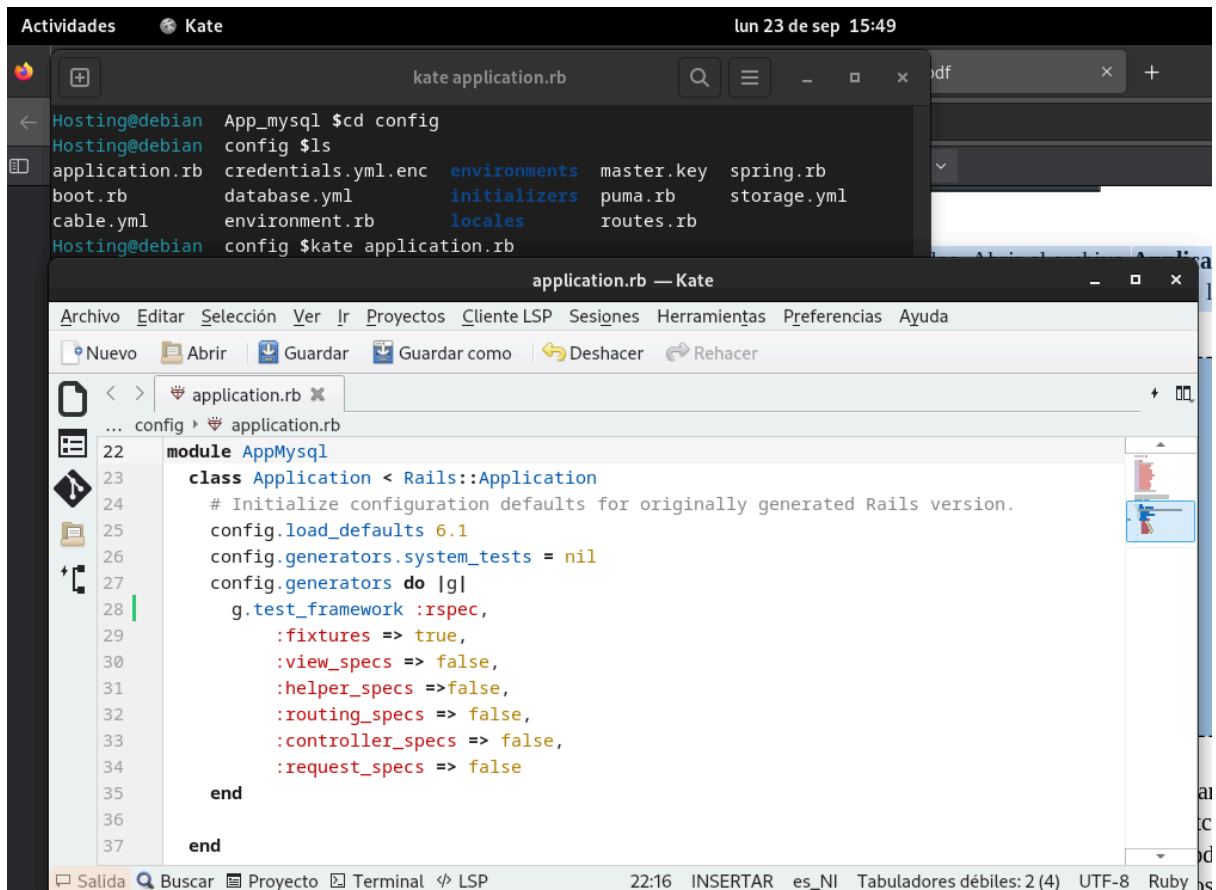
A screenshot of a code editor window titled 'rails_helper.rb' with the path '~/Proyectos_RoR/App_mysql/spec'. The code inside is as follows:

```
# This file is copied to spec/ when you run 'rails generate rspec:install'
require 'spec_helper'
ENV['RAILS_ENV'] ||= 'test'
require_relative '../config/environment'
# Prevent database truncation if the environment is production
abort("The Rails environment is running in production mode!") if Rails.env.production?
require 'rspec/rails'
require "capybara/rspec"
require 'shoulda/matchers'

# Add additional requires below this line. Rails is not loaded until this point!

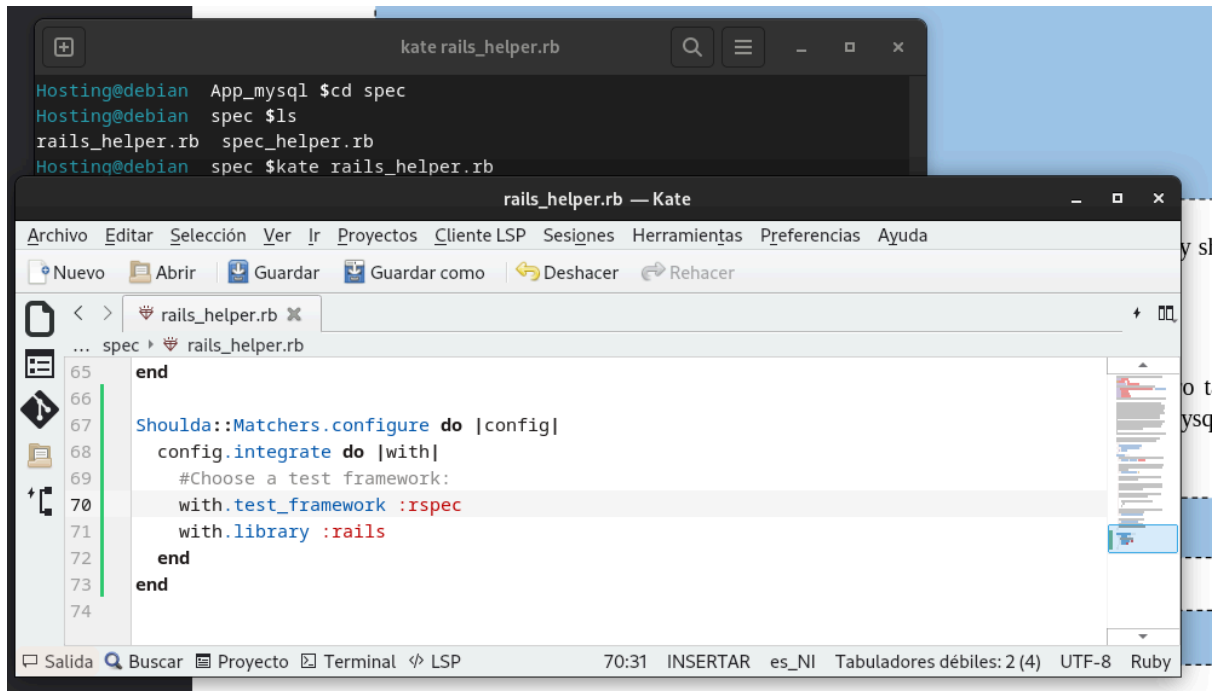
# Requires supporting ruby files with custom matchers and macros, etc, in
```

2.6 Como en esta guía solo se harán test para los modelos. Abrir el archivo Application.rb que está dentro del directorio config y agregar el siguiente código dentro de la clase applications

A screenshot of the Kate IDE interface. The top panel shows a file explorer with a tree view of the project files, including 'config' and 'application.rb'. The main editor window is titled 'application.rb — Kate' and shows the following code:

```
module AppMysql
  class Application < Rails::Application
    # Initialize configuration defaults for originally generated Rails version.
    config.load_defaults 6.1
    config.generators.system_tests = nil
    config.generators do |g|
      g.test_framework :rspec,
        :fixtures => true,
        :view_specs => false,
        :helper_specs => false,
        :routing_specs => false,
        :controller_specs => false,
        :request_specs => false
    end
  end
end
```

2.7 Por último solo queda escribir en siguiente código en el archivo /spec/rails_helper.rb para poder trabajar con la gema shoulda-matcher al momento de crear los tests. Agregarlo al final de todo el código que contiene el archivo.



The image shows a terminal window and a code editor window. The terminal window displays the following commands and output:

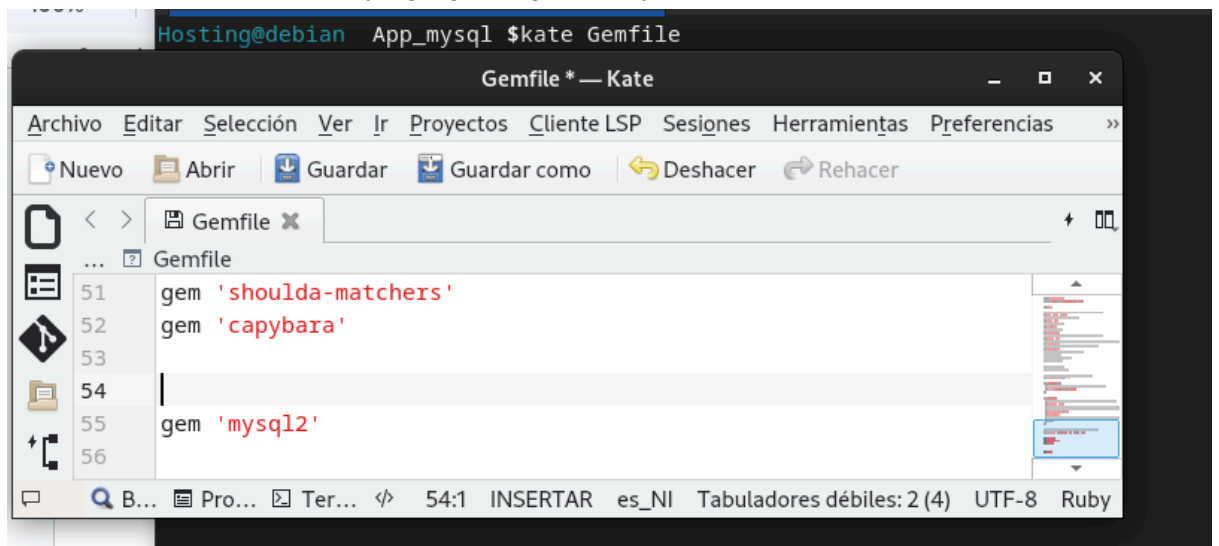
```
Hosting@debian App_mysql $cd spec
Hosting@debian spec $ls
rails_helper.rb spec_helper.rb
Hosting@debian spec $kate rails_helper.rb
```

The code editor window, titled "rails_helper.rb — Kate", shows the following code:

```
65 end
66
67 Shoulda::Matchers.configure do |config|
68   config.integrate do |with|
69     #Choose a test framework:
70     with.test_framework :rspec
71     with.library :rails
72   end
73 end
74
```

3.Trabajando con mysql

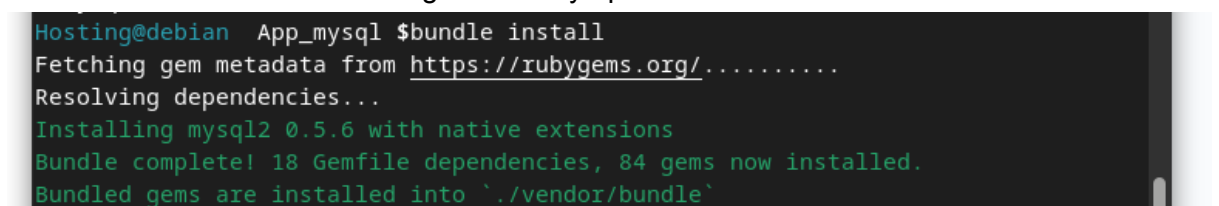
3.1 Abrir el archivo Gemfile y agregar la gema 'mysql2'.



The image shows a code editor window titled "Gemfile * — Kate". The code in the Gemfile is as follows:

```
51 gem 'shoulda-matchers'
52 gem 'capybara'
53
54 |
55 gem 'mysql2'
56
```

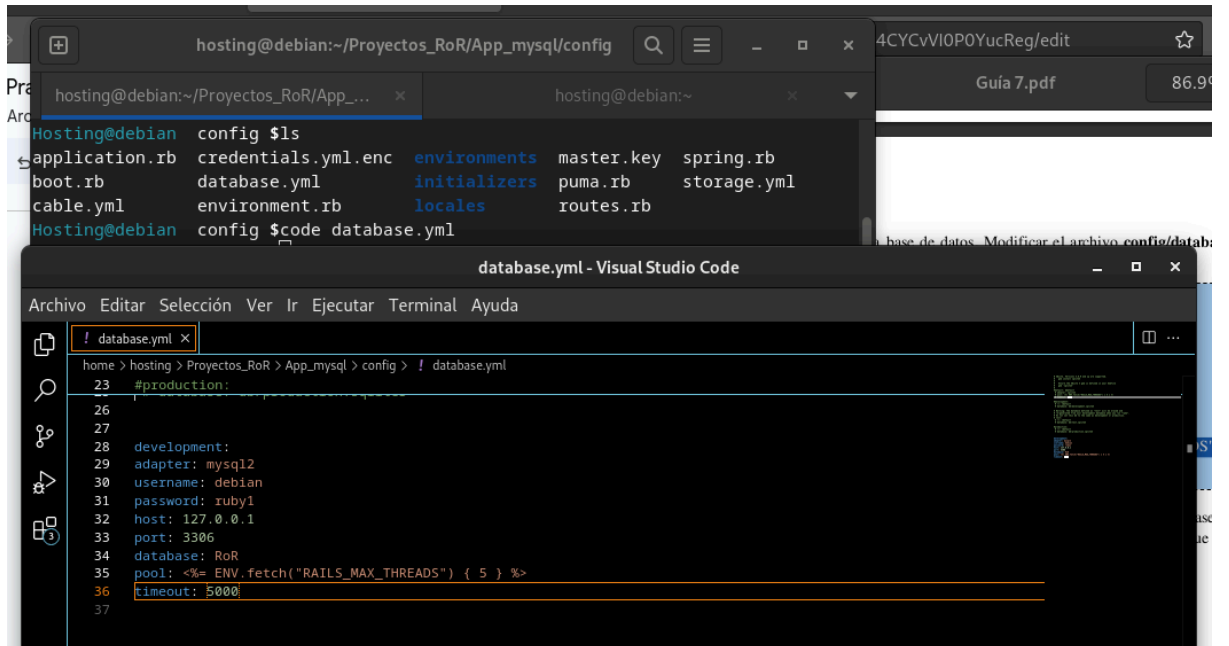
3.2 Desde el terminal instalar la gema de mysql.



The image shows a terminal window with the following output:

```
Hosting@debian App_mysql $bundle install
Fetching gem metadata from https://rubygems.org/.....
Resolving dependencies...
Installing mysql2 0.5.6 with native extensions
Bundle complete! 18 Gemfile dependencies, 84 gems now installed.
Bundled gems are installed into `./vendor/bundle`
```

4. Configurar la base de datos. Modificar el archivo config/database.yml para que quede de la siguiente manera, en la sección development.



The image shows a terminal window and a Visual Studio Code editor. The terminal window displays the output of the `ls` command in the `config` directory, listing files like `application.rb`, `boot.rb`, `credentials.yml.enc`, `database.yml`, `cable.yml`, `environment.rb`, `environments`, `initializers`, `locales`, `master.key`, `puma.rb`, `routes.rb`, `spring.rb`, and `storage.yml`. The Visual Studio Code editor shows the `database.yml` file with the following content:

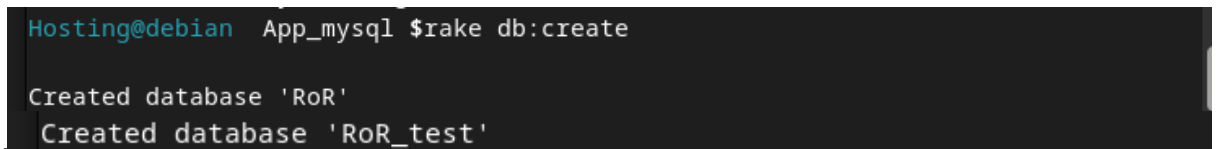
```
23 #production:
24
25 #development:
26
27
28 development:
29   adapter: mysql2
30   username: debian
31   password: ruby1
32   host: 127.0.0.1
33   port: 3306
34   database: RoR
35   pool: <%= ENV.fetch("RAILS_MAX_THREADS") { 5 } %>
36   timeout: 5000
37
```

5. Crear base de datos

5.1 Generar la base de datos.

Se tiene que diferenciar que es lo que se quiere hacer, si crear una nueva base de datos o migrar los datos de un motor de base de datos a otro.

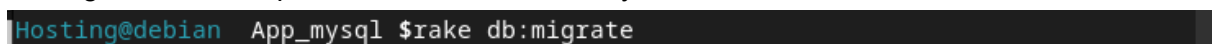
5.2 Crear nueva base de datos.



The terminal window shows the command `App_mysql $rake db:create` being executed. The output is:

```
Created database 'RoR'
Created database 'RoR_test'
```

5.3 Migrar los datos, para ello la base de datos ya debe existir.



The terminal window shows the command `App_mysql $rake db:migrate` being executed.

6. Para probar el comportamiento se está usando scaffold para generar código.

```
Hosting@debian App_mysql $rails g scaffold User name:string password:string email:string programador:boolean birthday:date work_time:time
Running via Spring preloader in process 8878
  invoke  active_record
  create  db/migrate/20240924231000_create_users.rb
  create  app/models/user.rb
  invoke  rspec
  create  spec/models/user_spec.rb
  invoke  resource_route
  route   resources :users
  invoke  scaffold_controller
  create  app/controllers/users_controller.rb
  invoke  erb
  create  app/views/users
  create  app/views/users/index.html.erb
  create  app/views/users/edit.html.erb
  create  app/views/users/show.html.erb
  create  app/views/users/new.html.erb
  create  app/views/users/_form.html.erb
  invoke  resource_route
  invoke  rspec
  invoke  helper
  create  app/helpers/users_helper.rb
  invoke  rspec
  invoke  jbuilder
  create  app/views/users/index.json.jbuilder
  create  app/views/users/show.json.jbuilder
  create  app/views/users/_user.json.jbuilder
  invoke  assets
  invoke  scss
  create  app/assets/stylesheets/users.scss
```

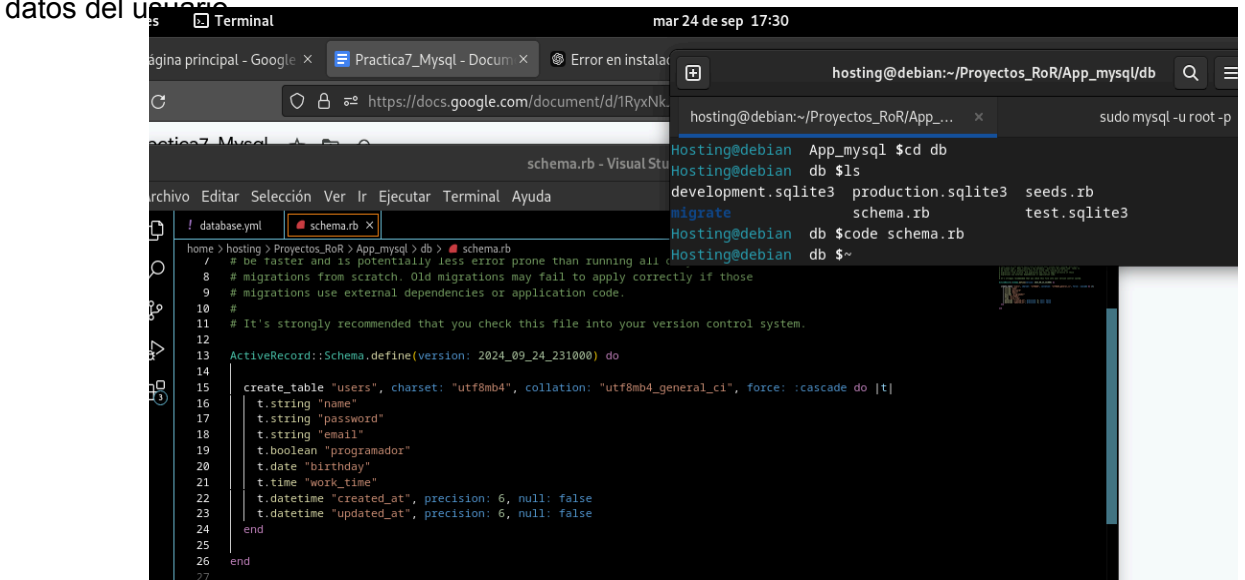
7. Migrar los datos generados con scaffold y la base de datos de los tests.

```
Hosting@debian App_mysql $rails db:migrate RAILS_ENV=test
Running via Spring preloader in process 9184
== 20240924231000 CreateUsers: migrating =====
-- create_table(:users)
   -> 0.0293s
== 20240924231000 CreateUsers: migrated (0.0296s) =====
```

8. Migrar los datos de la parte de desarrollo.

```
Hosting@debian App_mysql $rails db:migrate RAILS_ENV=development
Running via Spring preloader in process 9280
== 20240924231000 CreateUsers: migrating =====
-- create_table(:users)
   -> 0.0334s
== 20240924231000 CreateUsers: migrated (0.0337s) =====
```

9. Ir al directorio db y abrir el archivo schema.rb que se ha generado anteriormente, el cual contiene el esquema actual de la base de datos donde se almacenarán los datos del usuario.

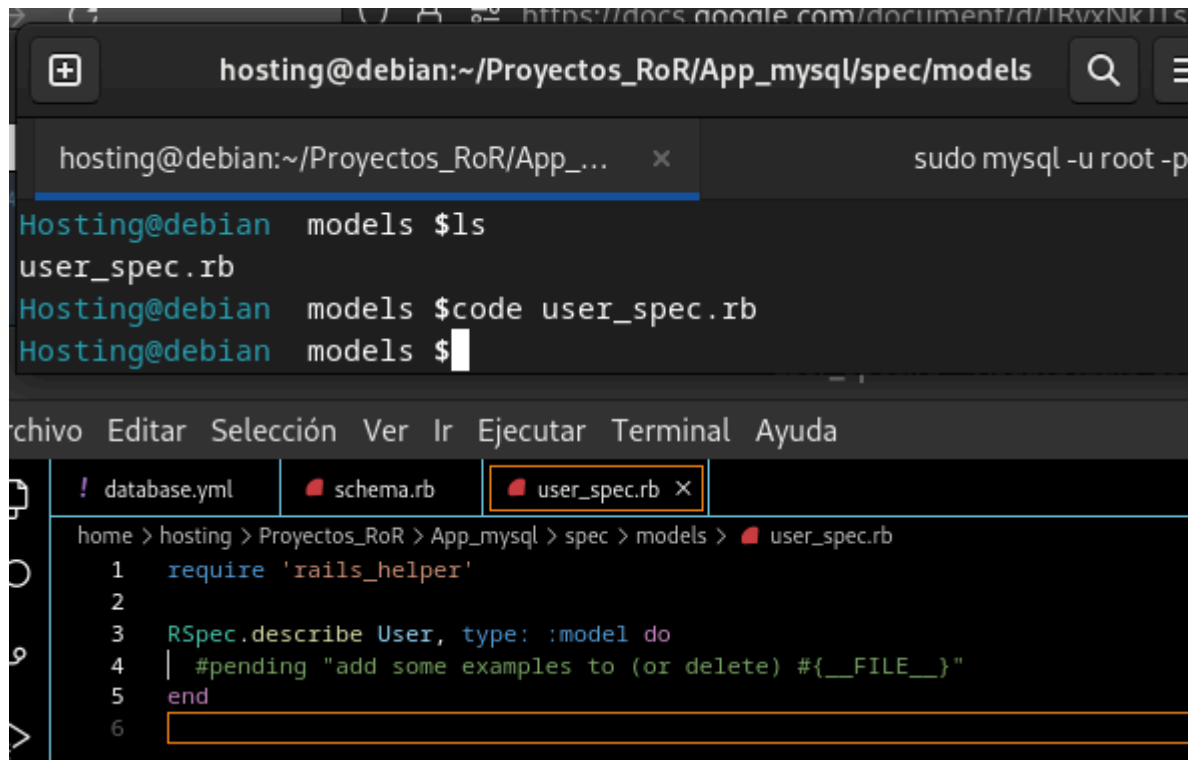


```
home > hosting > Proyectos_RoR > App_mysql > db > schema.rb
# be faster and is potentially less error prone than running all
# migrations from scratch. Old migrations may fail to apply correctly if those
# migrations use external dependencies or application code.
# It's strongly recommended that you check this file into your version control system.
ActiveRecord::Schema.define(version: 2024_09_24_231000) do
  create_table "users", charset: "utf8mb4", collation: "utf8mb4_general_ci", force: :cascade do |t|
    t.string "name"
    t.string "password"
    t.string "email"
    t.boolean "programador"
    t.date "birthday"
    t.time "work_time"
    t.datetime "created_at", precision: 6, null: false
    t.datetime "updated_at", precision: 6, null: false
  end
end
```

10. Si observa, al momento de generar el código con scaffold, se ha creado un directorio models dentro del directorio spec, verificar como se muestra a continuación.

```
Hosting@debian App_mysql $ ls spec/  
models rails_helper.rb spec_helper.rb
```

10.1 Dentro de este directorio model se ha creado el archivo user_spec.rb que se necesita para poder hacer test a los modelos. Eliminar o comentar la siguiente línea, que no se ocupará.



The screenshot shows a terminal window with the following commands and output:

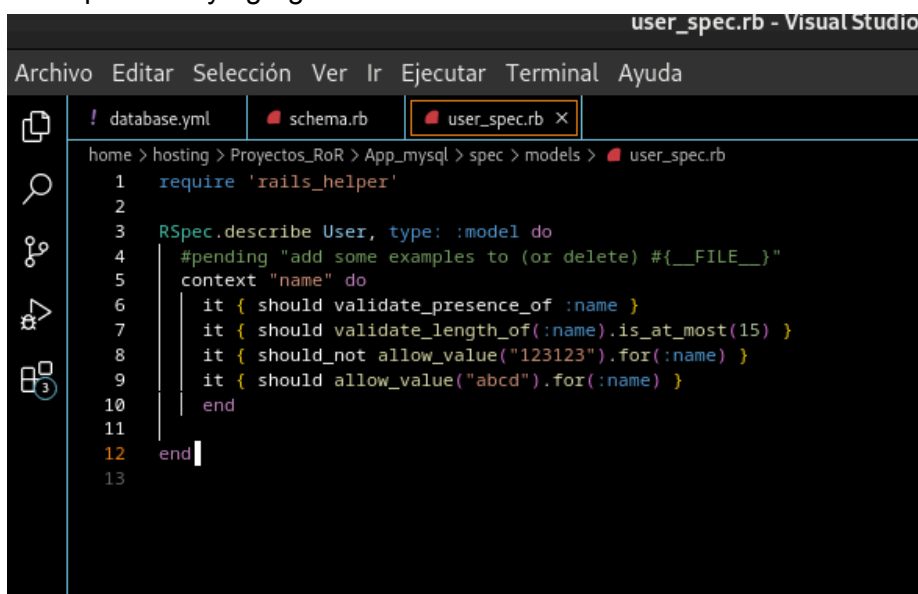
```
Hosting@debian:~/Proyectos_RoR/App_mysql/spec/models  
Hosting@debian models $ls  
user_spec.rb  
Hosting@debian models $code user_spec.rb  
Hosting@debian models $
```

Below the terminal, a code editor window titled "user_spec.rb" is open, showing the following code:

```
1 require 'rails_helper'  
2  
3 RSpec.describe User, type: :model do  
4   #pending "add some examples to (or delete) #{__FILE__}"  
5 end  
6
```

11. Para empezar a crear los tests y validar el modelo user.

11.1 Agregar el código al archivo spec/models/user_spec.rb para hacer test al campo name y agregar validaciones.



The screenshot shows a code editor window titled "user_spec.rb - Visual Studio" with the following code:

```
1 require 'rails_helper'  
2  
3 RSpec.describe User, type: :model do  
4   #pending "add some examples to (or delete) #{__FILE__}"  
5   context "name" do  
6     it { should validate_presence_of :name }  
7     it { should validate_length_of(:name).is_at_most(15) }  
8     it { should_not allow_value("123123").for(:name) }  
9     it { should allow_value("abcd").for(:name) }  
10  end  
11  
12 end  
13
```

11.2 Ir al terminal y ejecutar el siguiente comando para hacer los tests.

```
hosting@debian:~/Proyectos_RoR/App_mysql

hosting@debian:~/Proyectos_RoR/App_mysql
Hosting@debian App_mysql $bundle exec rspec spec/models/user_spec.rb
Running via Spring preloader in process 9978
FFF.

Failures:

  1) User name is expected to validate that :name cannot be empty/falsy
     Failure/Error: it { should validate_presence_of :name }

     Expected User to validate that :name cannot be empty/falsy, but this
     could not be proved.
     After setting :name to <">, the matcher expected the User to be
     invalid, but it was valid instead.
     # ./spec/models/user_spec.rb:6:in `block (3 levels) in <top (required)>'

  2) User name is expected to validate that the length of :name is at most 15
     Failure/Error: it { should validate_length_of(:name).is_at_most(15) }

     Expected User to validate that the length of :name is at most 15, but
     this could not be proved.
     After setting :name to <"xxxxxxxxxxxxxxxx">, the matcher expected the
     User to be invalid, but it was valid instead.
     # ./spec/models/user_spec.rb:7:in `block (3 levels) in <top (required)>'

  3) User name is expected not to allow :name to be <"123123">
     Failure/Error: it { should_not allow_value("123123").for(:name) }

     After setting :name to <"123123">, the matcher expected the User to be
     invalid, but it was valid instead.
     # ./spec/models/user_spec.rb:8:in `block (3 levels) in <top (required)>'
```

12. Ir al archivo user.rb que se encuentra en el directorio app/model y agregar el siguiente código dentro de la clase.

```
hosting@debian:~/Proyectos_RoR/App_mysql/app/models

Hosting@debian App_mysql $cd app/models
Hosting@debian models $code user.rb
Hosting@debian models $

user.rb - Visual Studio Code

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
! database.yml user.rb schema.rb user_spec.rb
home > hosting > Proyectos_RoR > App_mysql > app > models > user.rb
1 class User < ApplicationRecord
2   validates_presence_of :name, presence: true
3   validates_format_of :name, :with => /\A[a-zA-Z]+\z/
4   validates :name, length: { maximum: 15 }, on: :create
5 end
6
```

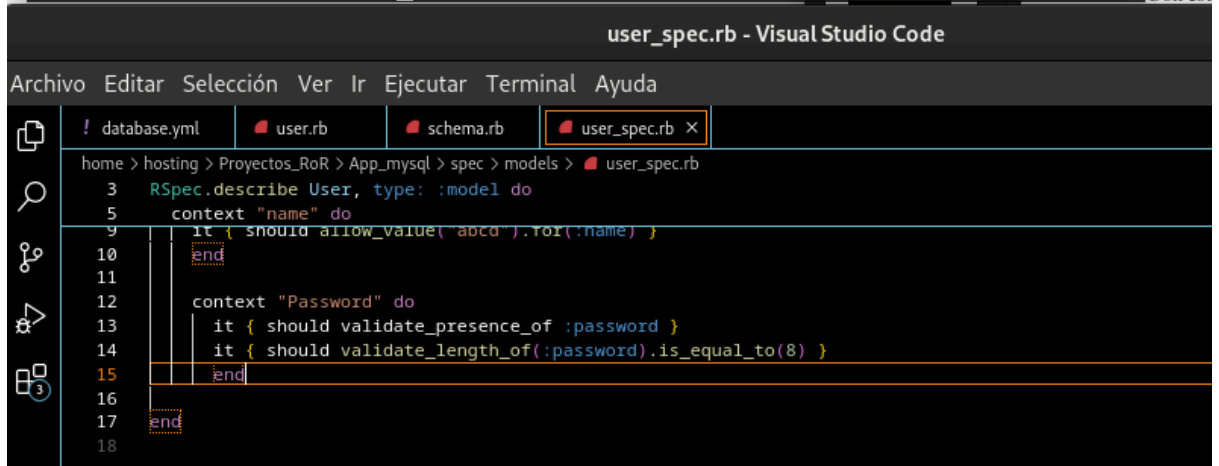

12.1 Si ejecuta el comando de los tests nuevamente, podrá observar que ya no se muestran los mensajes en rojo.

```
Hosting@debian App_mysql $bundle exec rspec spec/models/user_spec.rb
....

Finished in 0.65339 seconds (files took 9.89 seconds to load)
4 examples, 0 failures
```

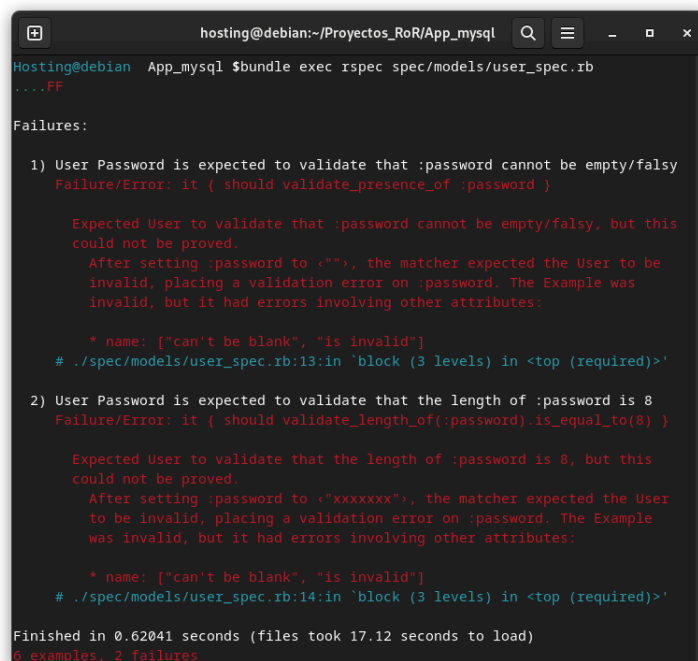
13. Agregar siguiente código al archivo spec/models/user_spec.rb para hacer test al campo password del modelo.

```
Hosting@debian App_mysql $code spec/models/user_spec.rb
Hosting@debian App_mysql $
```



```
user_spec.rb - Visual Studio Code
Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
! database.yml user.rb schema.rb user_spec.rb x
home > hosting > Proyectos_RoR > App_mysql > spec > models > user_spec.rb
3 RSpec.describe User, type: :model do
5   context "name" do
9     it { should allow_value( 'abcd' ).for(:name) }
10   end
11
12   context "Password" do
13     it { should validate_presence_of :password }
14     it { should validate_length_of(:password).is_equal_to(8) }
15   end
16
17 end
18
```

Al ejecutar el test, mostrará nuevamente los mensajes en color rojo.



```
hosting@debian:~/Proyectos_RoR/App_mysql
Hosting@debian App_mysql $bundle exec rspec spec/models/user_spec.rb
....FF

Failures:

  1) User Password is expected to validate that :password cannot be empty/falsy
     Failure/Error: it { should validate_presence_of :password }

     Expected User to validate that :password cannot be empty/falsy, but this
     could not be proved.
     After setting :password to <"">, the matcher expected the User to be
     invalid, placing a validation error on :password. The Example was
     invalid, but it had errors involving other attributes:

       * name: ["can't be blank", "is invalid"]
     # ./spec/models/user_spec.rb:13:in `block (3 levels) in <top (required)>'

  2) User Password is expected to validate that the length of :password is 8
     Failure/Error: it { should validate_length_of(:password).is_equal_to(8) }

     Expected User to validate that the length of :password is 8, but this
     could not be proved.
     After setting :password to <"xxxxxxx">, the matcher expected the User
     to be invalid, placing a validation error on :password. The Example
     was invalid, but it had errors involving other attributes:

       * name: ["can't be blank", "is invalid"]
     # ./spec/models/user_spec.rb:14:in `block (3 levels) in <top (required)>'

Finished in 0.62041 seconds (files took 17.12 seconds to load)
6 examples, 2 failures
```

14. Agregar las validaciones en el archivo app/model/user.rb, correspondientes para el campo password del modelo.

```
5
6   #para password
7   validates_presence_of :password, presence: true
8   validates_length_of :password, is: 8
9
10  end
```

14.1 Una vez escrito el código, volver a ejecutar el test para poder observar que ahora muestra el mensaje en verde, indicando que ha pasado los test a como se muestra en la siguiente figura.

```
Hosting@debian App_mysql $bundle exec rspec spec/models/user_spec.rb
.....

Finished in 0.29289 seconds (files took 13.1 seconds to load)
6 examples, 0 failures
```

15. El último test será para el campo email del modelo, agregar el código al archivo spec/models/user_spec.rb.

```
16
17   context "email" do
18     it { should validate_presence_of :email }
19     it { should validate_uniqueness_of :email }
20     it { should_not allow_value("text").for(:email) }
21     it { should allow_value("a@b.com").for(:email) }
22   end
23
24 end
```

16. Ejecutar el test en el terminal y mostrará los mensajes en rojo nuevamente.

```
2) User email is expected to validate that :email is case-sensitively unique
Failure/Error: it { should validate_uniqueness_of :email }

Expected User to validate that :email is case-sensitively unique, but
this could not be proved.
After taking the given User, setting its :email to <"dummy value">,
and saving it as the existing record, then making a new User and
setting its :email to <"dummy value"> as well, the matcher expected
the new User to be invalid, placing a validation error on :email. The
Example was invalid, but it had errors involving other attributes:

* name: ["can't be blank", "is invalid"]
* password: ["can't be blank", "is the wrong length (should be 8
characters)"]
# ./spec/models/user_spec.rb:19:in `block (3 levels) in <top (required)>'

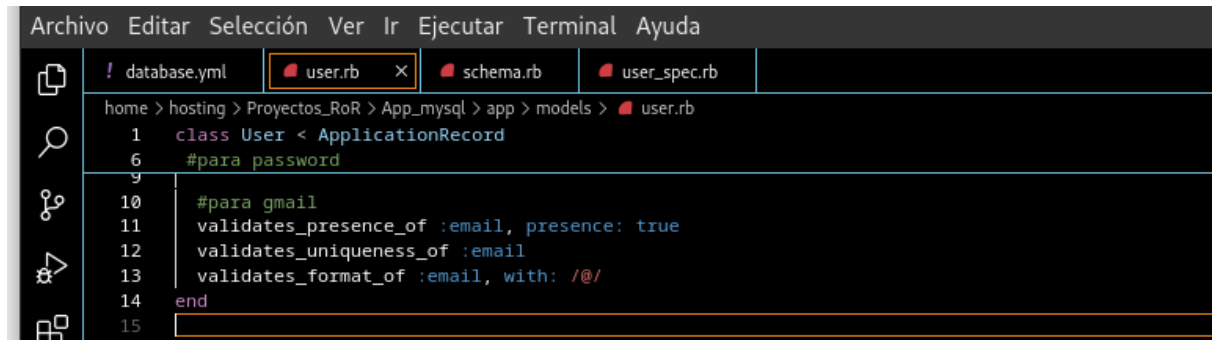
3) User email is expected not to allow :email to be <"text">
Failure/Error: it { should_not allow_value("text").for(:email) }

After setting :email to <"text">, the matcher expected the User to be
invalid, placing a validation error on :email. The Example was invalid,
but it had errors involving other attributes:

* name: ["can't be blank", "is invalid"]
* password: ["can't be blank", "is the wrong length (should be 8
characters)"]
# ./spec/models/user_spec.rb:20:in `block (3 levels) in <top (required)>'

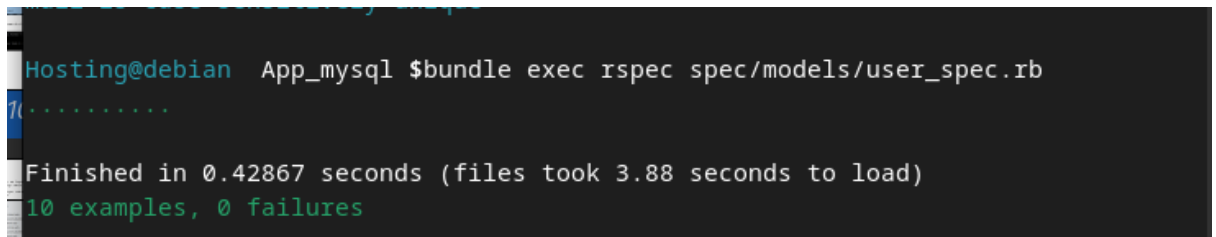
Finished in 0.41592 seconds (files took 7.07 seconds to load)
10 examples, 3 failures
```

17. Agregar las validaciones al modelo user en el archivo app/model/user.rb.



```
Archivo  Editar  Selección  Ver  Ir  Ejecutar  Terminal  Ayuda
! database.yml  user.rb  schema.rb  user_spec.rb
home > hosting > Proyectos_RoR > App_mysql > app > models > user.rb
1  class User < ApplicationRecord
6    #para password
9
10   #para gmail
11   validates_presence_of :email, presence: true
12   validates_uniqueness_of :email
13   validates_format_of :email, with: /@/
14 end
15
```

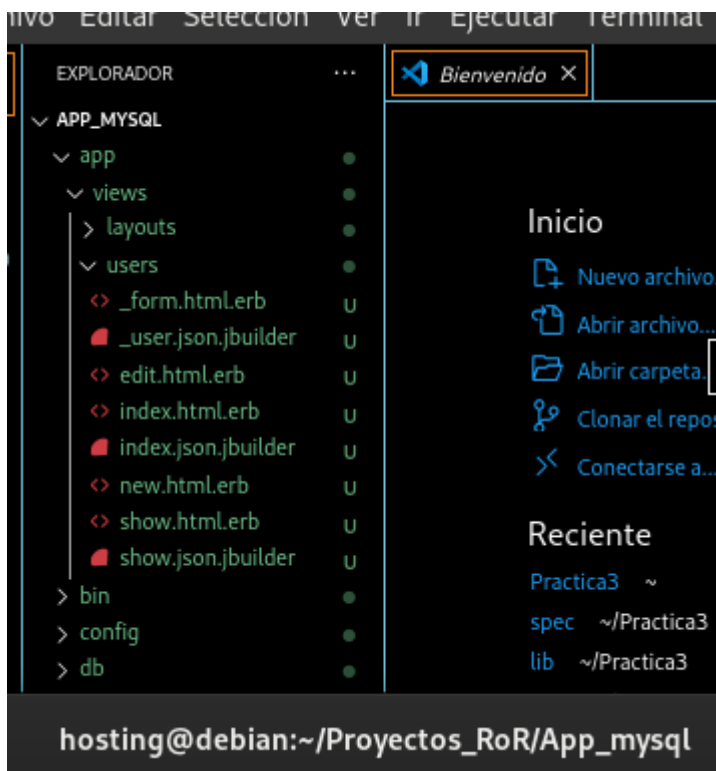
Ejecutar los tests y observar que finalmente todos los tests pasan y ya se han validado los campos del modelo user.



```
Hosting@debian App_mysql $bundle exec rspec spec/models/user_spec.rb
1/.....
Finished in 0.42867 seconds (files took 3.88 seconds to load)
10 examples, 0 failures
```

18. Hasta el momento no se ha trabajado con la parte web del framework; como se sabe, cuando se utiliza scaffold para generar el código, este ya ha generado las vistas para el modelo user.

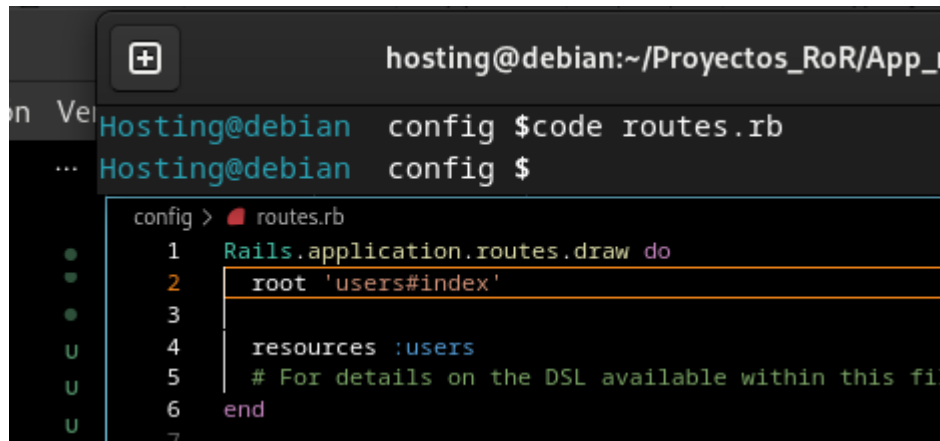
18.1 Ir al directorio view y observar que existen todas las vistas necesarias para el correcto funcionamiento del modelo user.



```
EXPLORADOR  Bienvenido
APP_MYSQL
└─ app
   └─ views
      ├── layouts
      └─ users
         ├── _form.html.erb
         ├── _user.json.jbuilder
         ├── edit.html.erb
         ├── index.html.erb
         ├── index.json.jbuilder
         ├── new.html.erb
         ├── show.html.erb
         └── show.json.jbuilder
└─ bin
└─ config
└─ db
```

hosting@debian:~/Proyectos_RoR/App_mysql

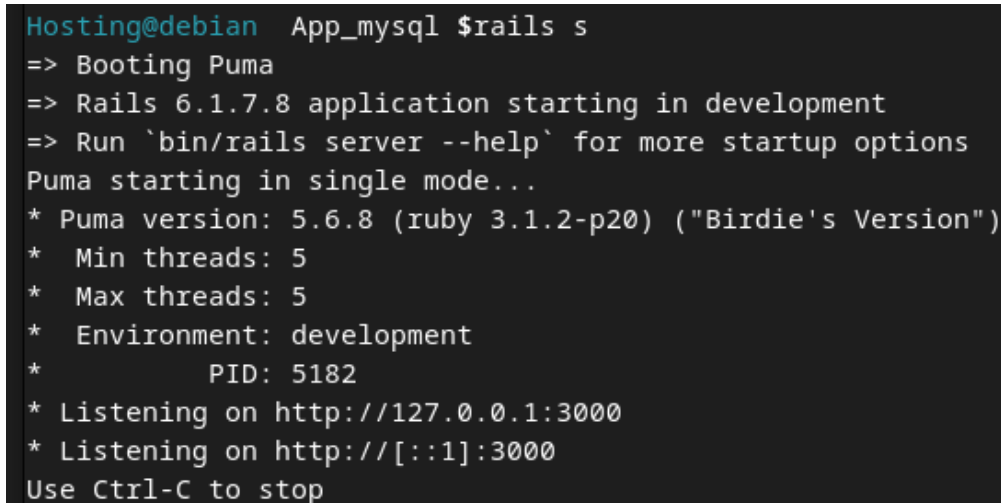
18.2 El siguiente paso es configurar el index de la aplicación, para eso ir al archivo config/routes.rb y escribir el código para el index sea la página principal del proyecto. El archivo routes quedará como se muestra en la figura 82.



```
hosting@debian:~/Proyectos_RoR/App_r
Hosting@debian config $code routes.rb
... Hosting@debian config $

config > routes.rb
1 Rails.application.routes.draw do
2   root 'users#index'
3
4   resources :users
5   # For details on the DSL available within this file
6 end
7
```

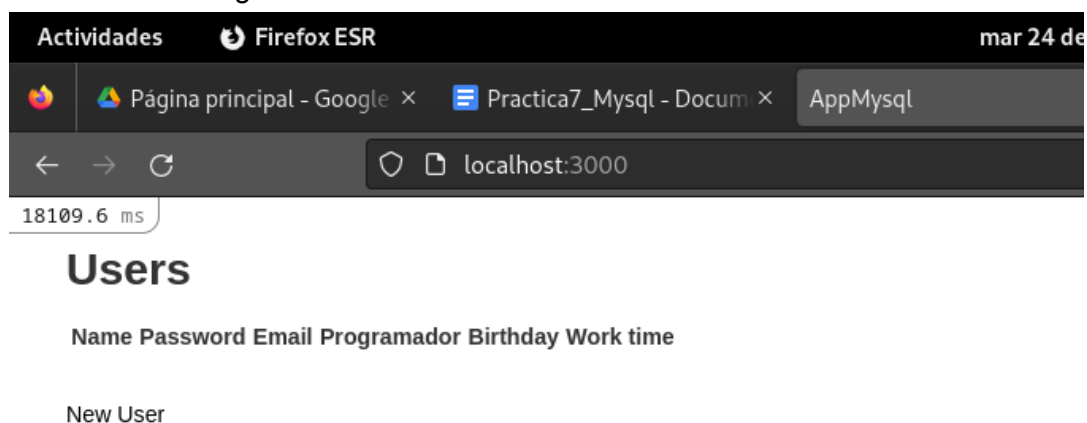
19. Para probar la aplicación, debe arrancar el servidor en el terminal, ejecutando el siguiente comando.



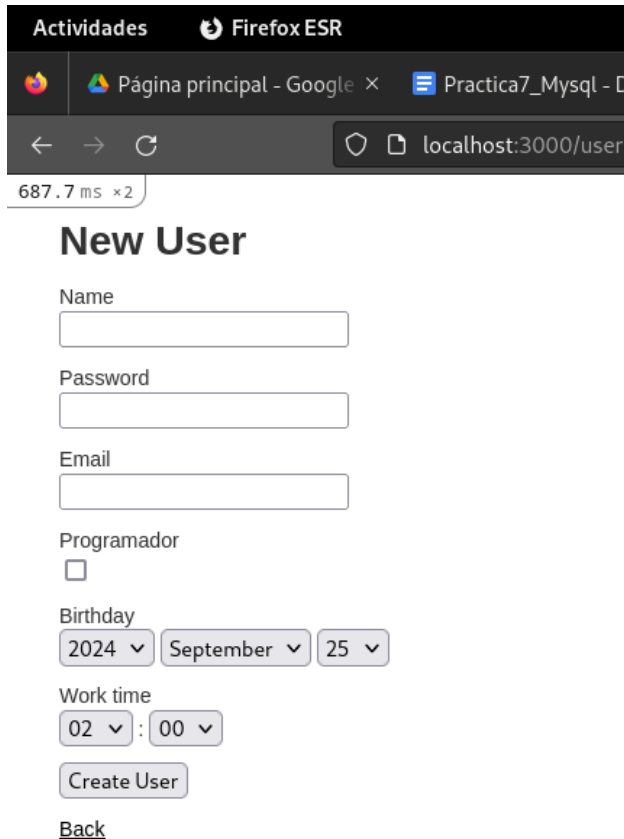
```
Hosting@debian App_mysql $rails s
=> Booting Puma
=> Rails 6.1.7.8 application starting in development
=> Run `bin/rails server --help` for more startup options
Puma starting in single mode...
* Puma version: 5.6.8 (ruby 3.1.2-p20) ("Birdie's Version")
* Min threads: 5
* Max threads: 5
* Environment: development
* PID: 5182
* Listening on http://127.0.0.1:3000
* Listening on http://[::1]:3000
Use Ctrl-C to stop
```

20. Prueba en la aplicación web

20.1 Abrir el navegador en la dirección localhost:3000



20.2 Para crear nuevo usuario y agregar datos, presionar click en New User y mostrará el formulario para ingresar los datos.



Actividades Firefox ESR

Página principal - Google × Practica7_Mysql - D

← → ↻ localhost:3000/users

687.7 ms ×2

New User

Name

Password

Email

Programador
☐

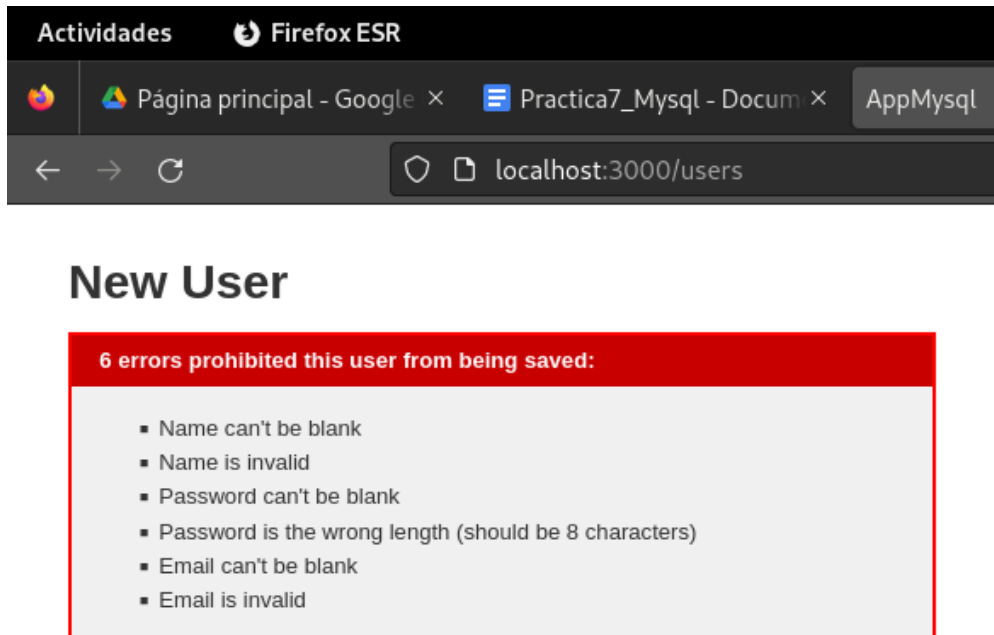
Birthday
2024 ▼ September ▼ 25 ▼

Work time
02 ▼ : 00 ▼

Create User

[Back](#)

20.3 Una vez abierto el formulario, se podrá poner a prueba las validaciones que se agregaron anteriormente, si presiona el botón create user observará que muestra unos mensajes en relación a las validaciones de los campos del modelo user. Estos mensajes son generados automáticamente por el framework.



Actividades Firefox ESR

Página principal - Google × Practica7_Mysql - Docum × AppMysql

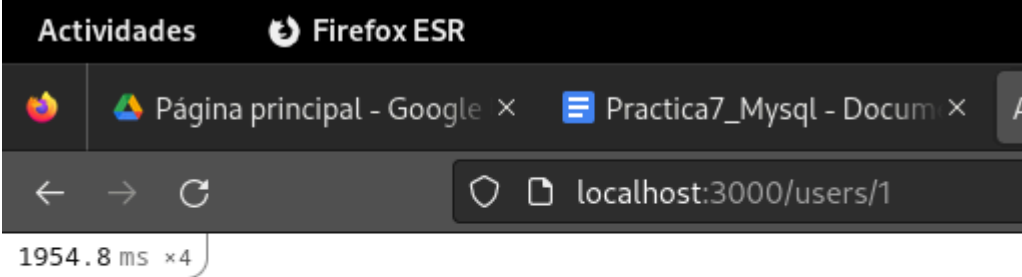
← → ↻ localhost:3000/users

New User

6 errors prohibited this user from being saved:

- Name can't be blank
- Name is invalid
- Password can't be blank
- Password is the wrong length (should be 8 characters)
- Email can't be blank
- Email is invalid

20.4 Al llenar los datos del formulario correctamente, ya permite guardar los datos.



The screenshot shows a Firefox ESR browser window with the address bar at `localhost:3000/users/1`. Below the browser, a green message states "User was successfully created." followed by the user details:

Name: Juan
Password: 12345678
Email: juan@gmail.com
Programador: true
Birthday: 2019-05-22
Work time: 2000-01-01 13:29:00 UTC

At the bottom, there are links for [Edit](#) and [Back](#).

20.5 Verificar que se haya creado correctamente el usuario en mysql, abrir el phpmyadmin y hacer un select a la tabla user, de la base de datos en la que se está trabajando.

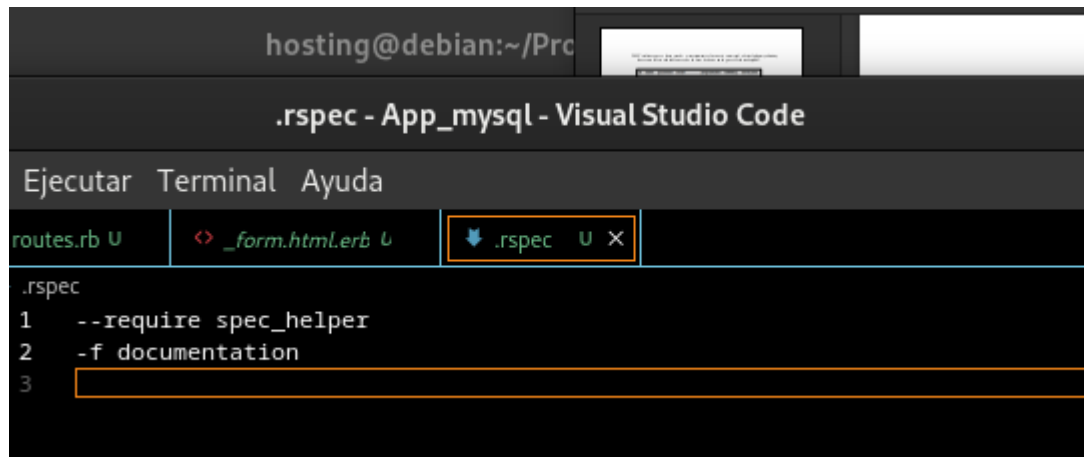
id	name	password	email	programador	birthday	work_time
1	Juan	12345678	juan@gmail.com	1	2019-05-22	13:29:00

21.Pruebas por comportamiento BDD.

21.1 Para hacer testing primero crear el directorio en donde se guardan los features.

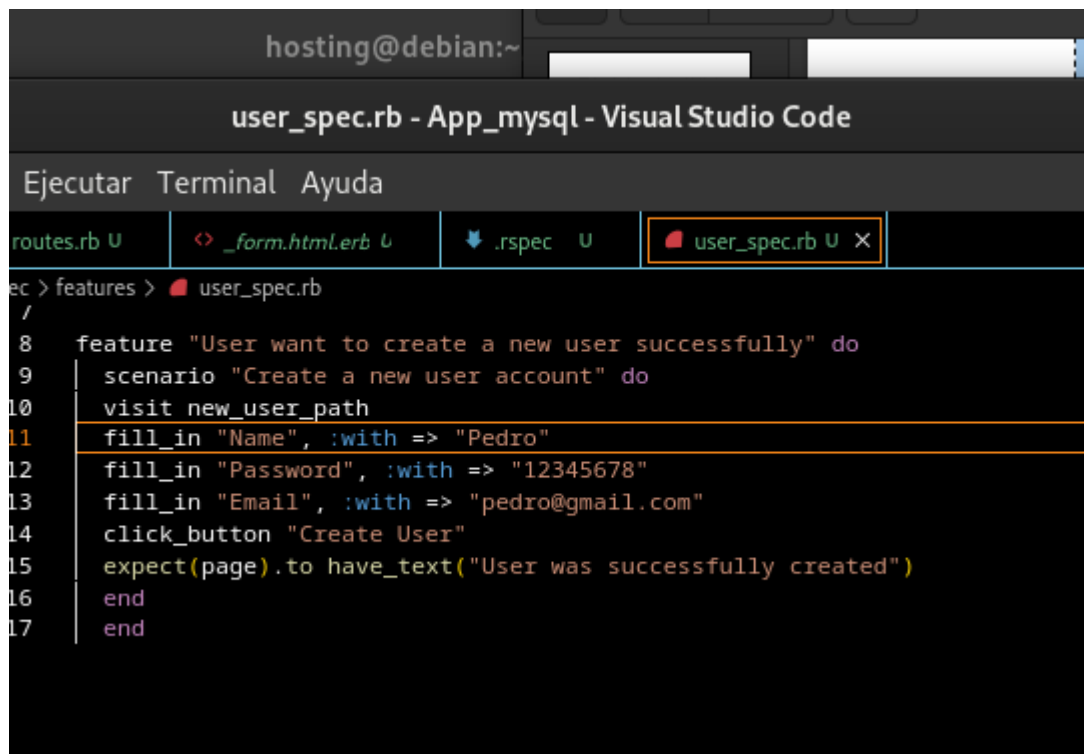
```
- Goodbye!  
Exiting  
Hosting@debian App_mysql $rails g rspec:feature user  
Running via Spring preloader in process 8072  
create spec/features/user_spec.rb
```

21.2 Abrir el archivo `.rspec` que está en la raíz de la aplicación y agregar la siguiente línea.



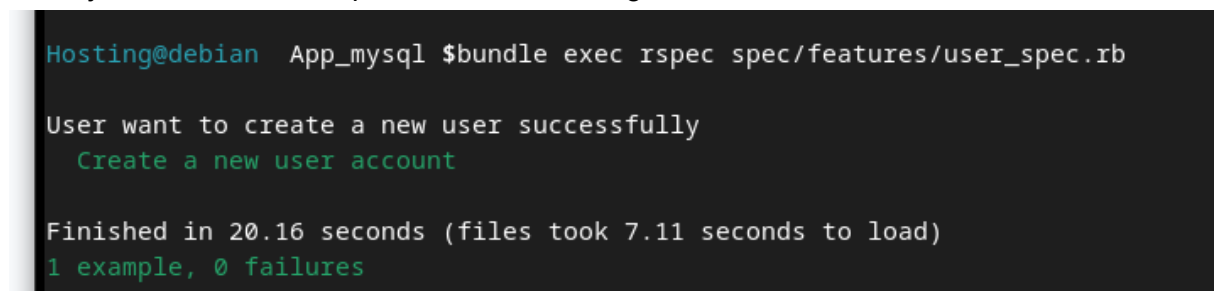
```
hosting@debian:~/Pro  
.rspec - App_mysql - Visual Studio Code  
Ejecutar Terminal Ayuda  
routes.rb U _form.html.erb U .rspec U X  
.rspec  
1 --require spec_helper  
2 -f documentation  
3
```

21.3 Abrir el archivo `spec/features/user_spec.rb` y agregar el siguiente código, para hacer test al comportamiento al momento de crear un nuevo usuario.



```
hosting@debian:~  
user_spec.rb - App_mysql - Visual Studio Code  
Ejecutar Terminal Ayuda  
routes.rb U _form.html.erb U .rspec U user_spec.rb U X  
ec > features > user_spec.rb  
/  
8 feature "User want to create a new user successfully" do  
9 | scenario "Create a new user account" do  
10 | visit new_user_path  
11 | fill_in "Name", :with => "Pedro"  
12 | fill_in "Password", :with => "12345678"  
13 | fill_in "Email", :with => "pedro@gmail.com"  
14 | click_button "Create User"  
15 | expect(page).to have_text("User was successfully created")  
16 | end  
17 | end
```

21.4 Ejecutar el test con `rspec`, escribiendo el siguiente comando en el terminal.



```
Hosting@debian App_mysql $bundle exec rspec spec/features/user_spec.rb  
  
User want to create a new user successfully  
  Create a new user account  
  
Finished in 20.16 seconds (files took 7.11 seconds to load)  
1 example, 0 failures
```

22. Otra prueba que se puede realizar en relación al comportamiento, es cuando el usuario presione click en el link back se debe mostrar la página principal de la aplicación donde se muestran todos los usuarios almacenados.

22.1 Copiar el siguiente código en el archivo spec/features/user_spec.rb

```
14  
15 feature "User want to see all users created" do  
16   scenario "User is on index page" do  
17     visit new_user_path  
18     click_link "Back"  
19     expect(page).to have_content("Users")  
20   end  
21 end
```

22.2 Ejecutar el test nuevamente y podrá observar que se ejecuta sin ningún problema.

```
Hosting@debian App_mysql $bundle exec rspec spec/features/user_spec.rb
```

```
User want to create a new user successfully
```

```
  Create a new user account
```

```
User want to see all users created
```

```
  User is on index page
```

```
Finished in 1.72 seconds (files took 10.77 seconds to load)
```

```
2 examples, 0 failures
```


Ejercicios propuestos para ser entregados al docente

2. Crear una aplicación en Rails utilizando mysql; configurar una base de datos de nombre Datos_estudiante y utilice scaffold para generar el código y crear una tabla Estudiante con los campos nombres, apellidos, carrera, carnet y validar todos los campos necesarios.

Extra: para las validaciones hacer uso de TDD, haciendo uso las distintas gemas anteriormente utilizadas en la guías. Y agregar código css a las vistas para obtener un mejor diseño.

1. Para crear una nueva base de datos agregaremos el siguiente código con “secondary:” estamos diciendo que esa base de datos es la que queremos crear ya que la “primary:” esta creada.

```
development:
  secondary:
    <<: *default
    username: hosting
    password: h1234
    host: 127.0.0.1
    port: 3306
    database: Datos_estudiamtes

test:
  secondary:
    <<: *default
    username: hosting
    password: h1234
    database: Datos_estudiantes_test
```

2. Ahora ejecutamos el rake db: create para poder crear la base de datos.

```
Hosting@debian App_mysql $rake db:create
Created database 'Datos_estudiamtes'
Created database 'Datos_estudiantes_test'
```

3. verificamos que se ha creado en phpmyadmin



5. Usamos Scanffold para generar el código y la tabla estudiantes

```
Hosting@debian App_mysql $rails generate scaffold Estudiante nombres:string ape
llidos:string carrera:string carnet:string

Running via Spring preloader in process 18824
  invoke  active_record
  create  db/migrate/20240925193509_create_estudiantes.rb
  create  app/models/estudiante.rb
  invoke  rspec
  create  spec/models/estudiante_spec.rb
  invoke  resource_route
   route  resources :estudiantes
  invoke  scaffold_controller
  create  app/controllers/estudiantes_controller.rb
  invoke  erb
  create  app/views/estudiantes
  create  app/views/estudiantes/index.html.erb
  create  app/views/estudiantes/edit.html.erb
  create  app/views/estudiantes/show.html.erb
  create  app/views/estudiantes/new.html.erb
  create  app/views/estudiantes/_form.html.erb
  invoke  resource_route
  invoke  rspec
  invoke  helper
  create  app/helpers/estudiantes_helper.rb
```

6. Migramos los datos

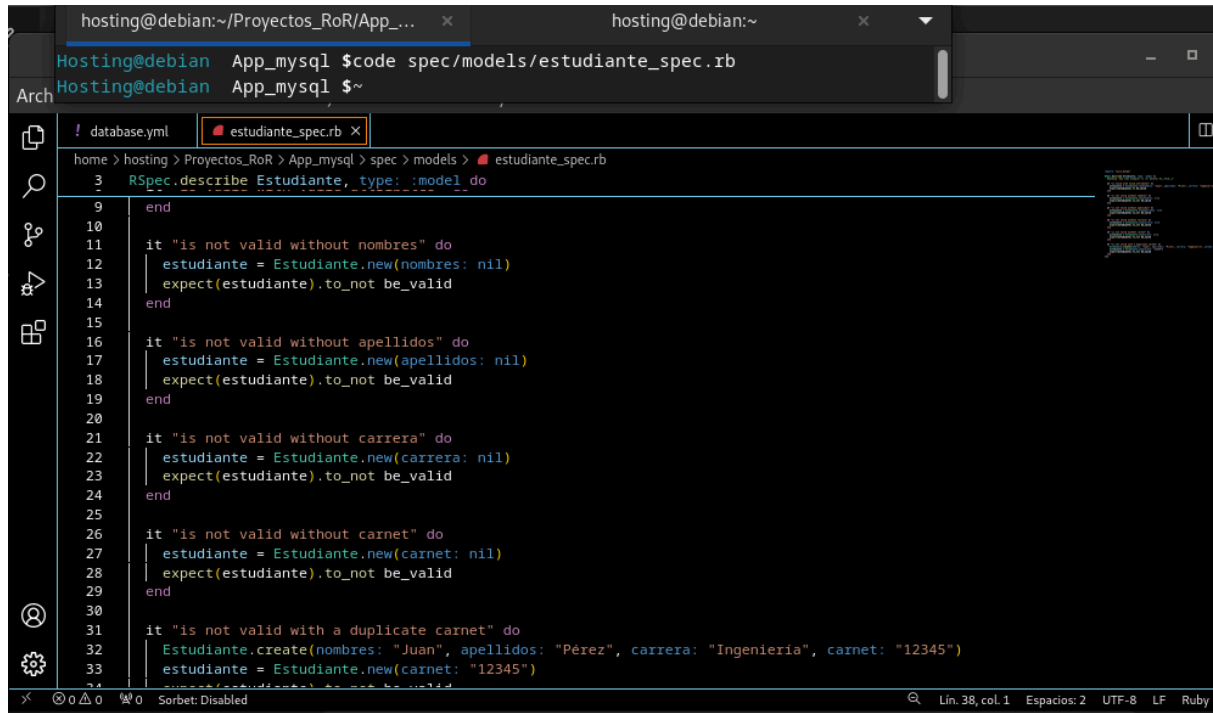
```
== 20240925193509 CreateEstudiantes: migrating =====
-- create_table(:estudiantes)
   -> 0.0171s
== 20240925193509 CreateEstudiantes: migrated (0.0243s) =====
```

7. ir al directorio db y abrir el archivo schema.rb que se ha generado anteriormente, el cual contiene el esquema actual de la base de datos donde se almacenarán los datos del usuario.

```
ActiveRecord::Schema.define(version: 2024_09_25_193509) do

  create_table "estudiantes", charset: "utf8mb4", collation: "utf8mb4_general_ci",
force: :cascade do |t|
    t.string "nombres"
    t.string "apellidos"
    t.string "carrera"
    t.string "carnet"
    t.datetime "created_at", precision: 6, null: false
    t.datetime "updated_at", precision: 6, null: false
  end
end
```

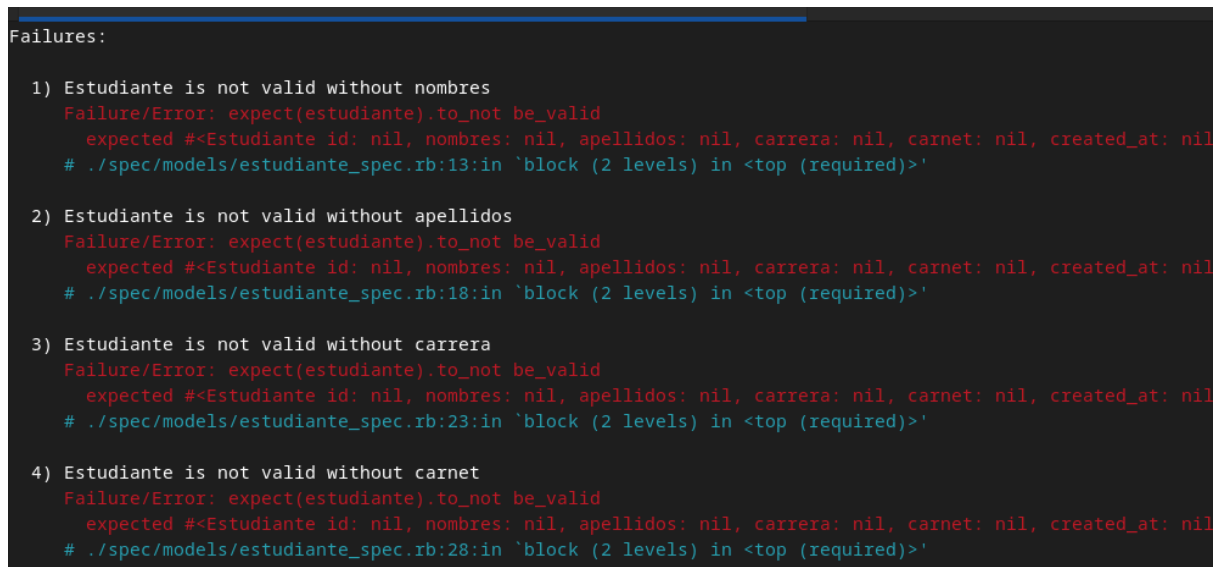
8. Agregar el código al archivo spec/models/user_spec.rb para hacer test a los campos y agregar validaciones.



```
hosting@debian:~/Proyectos_RoR/App... x hosting@debian:~ x
Hosting@debian App_mysql $code spec/models/estudiante_spec.rb
Hosting@debian App_mysql $~

! database.yml x estudiante_spec.rb x
home > hosting > Proyectos_RoR > App_mysql > spec > models > estudiante_spec.rb
3 RSpec.describe Estudiante, type: :model do
9   end
10
11   it "is not valid without nombres" do
12     estudiante = Estudiante.new(nombres: nil)
13     expect(estudiante).to_not be_valid
14   end
15
16   it "is not valid without apellidos" do
17     estudiante = Estudiante.new(apellidos: nil)
18     expect(estudiante).to_not be_valid
19   end
20
21   it "is not valid without carrera" do
22     estudiante = Estudiante.new(carrera: nil)
23     expect(estudiante).to_not be_valid
24   end
25
26   it "is not valid without carnet" do
27     estudiante = Estudiante.new(carnet: nil)
28     expect(estudiante).to_not be_valid
29   end
30
31   it "is not valid with a duplicate carnet" do
32     Estudiante.create(nombres: "Juan", apellidos: "Pérez", carrera: "Ingeniería", carnet: "12345")
33     estudiante = Estudiante.new(carnet: "12345")
34     expect(estudiante).to_not be_valid
35   end
end
```

9. Ir al terminal y ejecutar el siguiente comando para hacer los tests.



```
Failures:

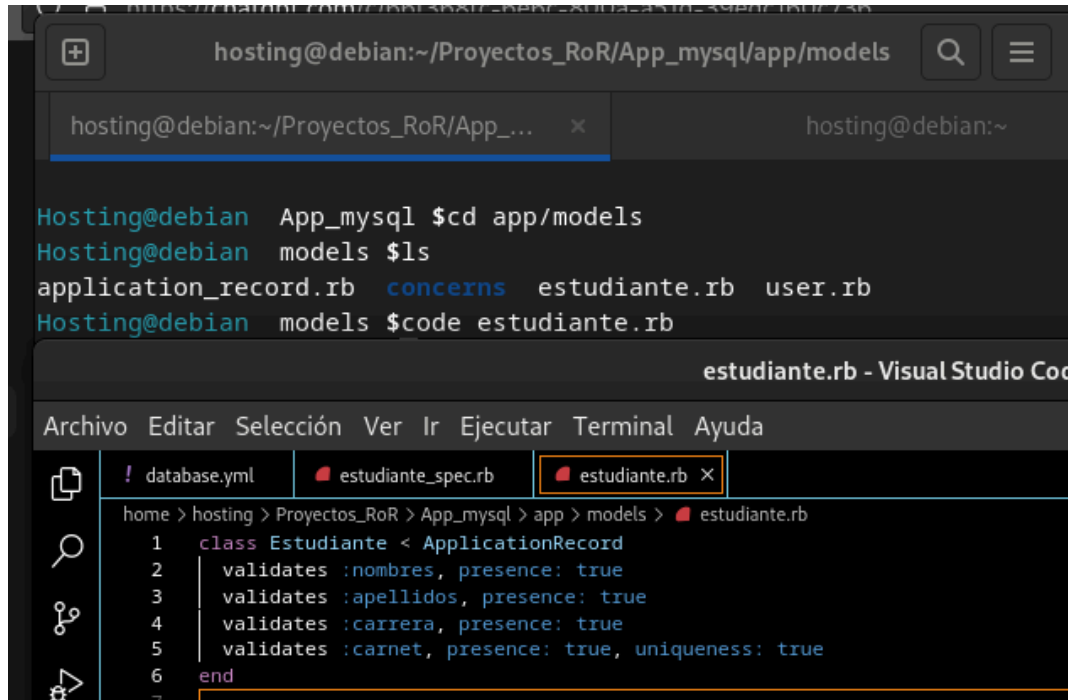
1) Estudiante is not valid without nombres
Failure/Error: expect(estudiante).to_not be_valid
expected #<Estudiante id: nil, nombres: nil, apellidos: nil, carrera: nil, carnet: nil, created_at: nil>
# ./spec/models/estudiante_spec.rb:13:in `block (2 levels) in <top (required)>'

2) Estudiante is not valid without apellidos
Failure/Error: expect(estudiante).to_not be_valid
expected #<Estudiante id: nil, nombres: nil, apellidos: nil, carrera: nil, carnet: nil, created_at: nil>
# ./spec/models/estudiante_spec.rb:18:in `block (2 levels) in <top (required)>'

3) Estudiante is not valid without carrera
Failure/Error: expect(estudiante).to_not be_valid
expected #<Estudiante id: nil, nombres: nil, apellidos: nil, carrera: nil, carnet: nil, created_at: nil>
# ./spec/models/estudiante_spec.rb:23:in `block (2 levels) in <top (required)>'

4) Estudiante is not valid without carnet
Failure/Error: expect(estudiante).to_not be_valid
expected #<Estudiante id: nil, nombres: nil, apellidos: nil, carrera: nil, carnet: nil, created_at: nil>
# ./spec/models/estudiante_spec.rb:28:in `block (2 levels) in <top (required)>'
```

10. Ir al archivo estudiante.rb que se encuentra en el directorio app/model y agregar el siguiente código dentro de la clase.



The screenshot shows a terminal window and the Visual Studio Code editor. The terminal shows the following commands and output:

```
Hosting@debian App_mysql $cd app/models
Hosting@debian models $ls
application_record.rb  concerns  estudiante.rb  user.rb
Hosting@debian models $code estudiante.rb
```

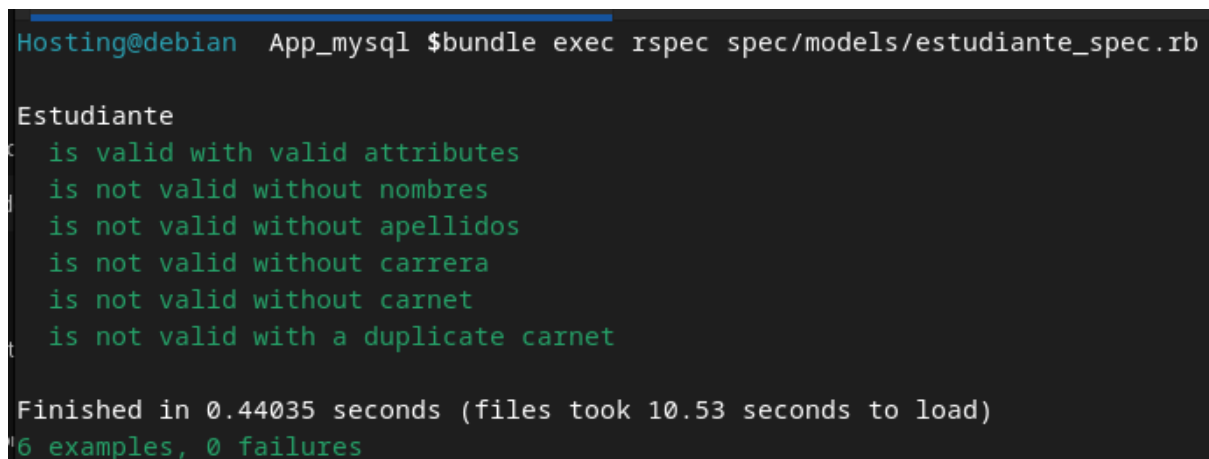
The Visual Studio Code editor shows the file explorer with the following structure:

- ! database.yml
- estudiante_spec.rb
- estudiante.rb (selected)

The code in estudiante.rb is as follows:

```
1 class Estudiante < ApplicationRecord
2   validates :nombres, presence: true
3   validates :apellidos, presence: true
4   validates :carrera, presence: true
5   validates :carnet, presence: true, uniqueness: true
6 end
```

11. Si ejecuta el comando de los tests nuevamente, podrá observar que ya no se muestran los mensajes en rojo.



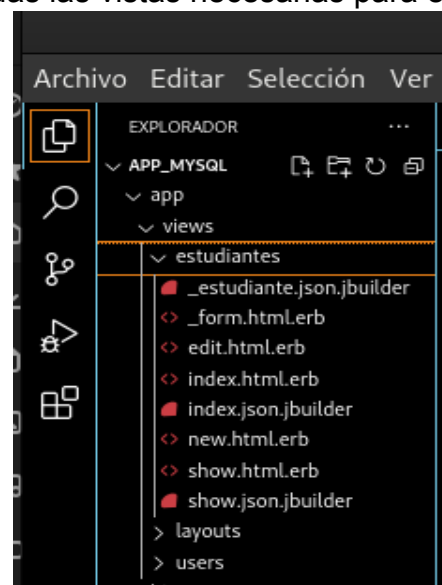
The screenshot shows a terminal window with the following output:

```
Hosting@debian App_mysql $bundle exec rspec spec/models/estudiante_spec.rb

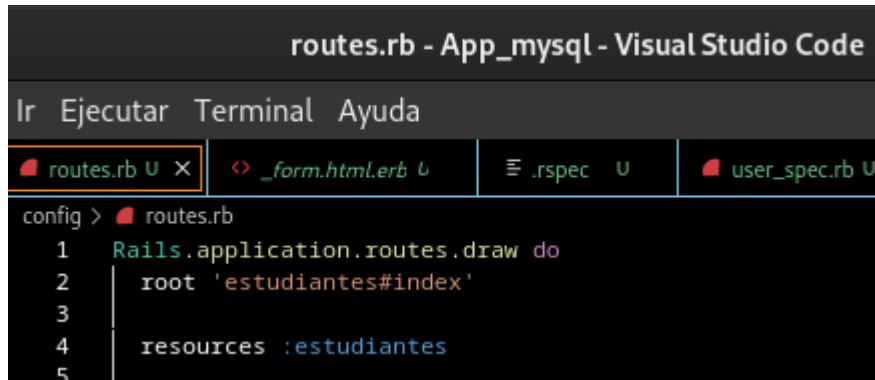
Estudiante
  is valid with valid attributes
  is not valid without nombres
  is not valid without apellidos
  is not valid without carrera
  is not valid without carnet
  is not valid with a duplicate carnet

Finished in 0.44035 seconds (files took 10.53 seconds to load)
6 examples, 0 failures
```

12. Ir al directorio view y observar que existen todas las vistas necesarias para el correcto funcionamiento del modelo estudiante.



13. El siguiente paso es configurar el index de la aplicación, para eso ir al archivo config/routes.rb y escribir el código para el index sea la página principal del proyecto. El archivo routes



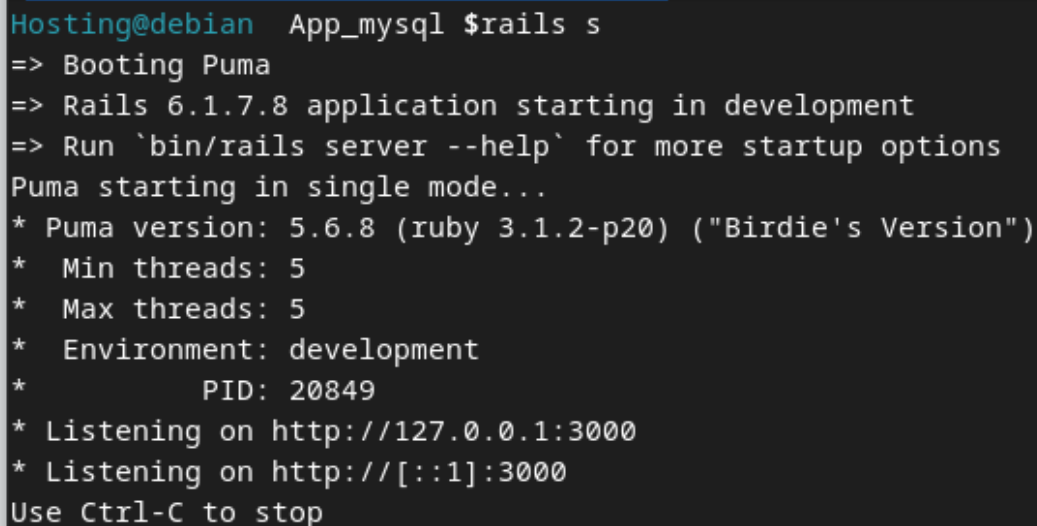
```
routes.rb - App_mysql - Visual Studio Code

Ir Ejecutar Terminal Ayuda

routes.rb x <> _form.html.erb u .rspec u user_spec.rb u

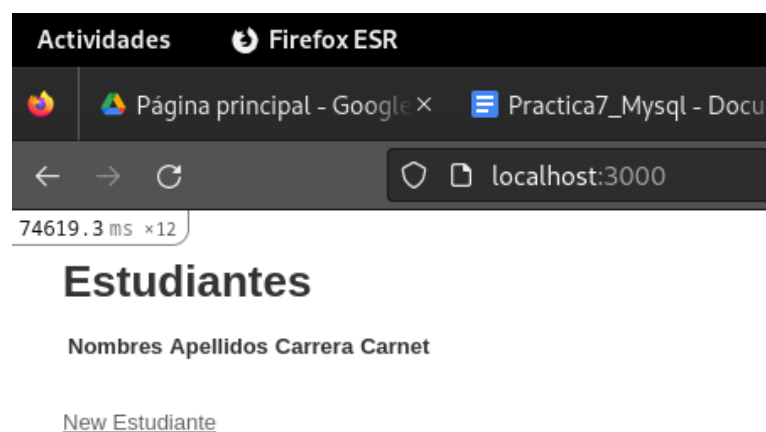
config > routes.rb
1 Rails.application.routes.draw do
2   root 'estudiantes#index'
3
4   resources :estudiantes
5
```

14. Para probar la aplicación, debe arrancar el servidor en el terminal, ejecutando el siguiente comando.



```
Hosting@debian App_mysql $ rails s
=> Booting Puma
=> Rails 6.1.7.8 application starting in development
=> Run `bin/rails server --help` for more startup options
Puma starting in single mode...
* Puma version: 5.6.8 (ruby 3.1.2-p20) ("Birdie's Version")
* Min threads: 5
* Max threads: 5
* Environment: development
* PID: 20849
* Listening on http://127.0.0.1:3000
* Listening on http://[::1]:3000
Use Ctrl-C to stop
```

15. Abrir el navegador en la dirección localhost:3000



16. Damos new estudiante y nos mostrara el formulario

848.7 ms

New Estudiante

Nombres

Apellidos

Carrera

Carnet

[Create Estudiante](#)

[Back](#)

17. Si damos en create estudiante no nos debe dejar por las validaciones

4 errors prohibited this estudiante from being saved:

- Nombres can't be blank
- Apellidos can't be blank
- Carrera can't be blank
- Carnet can't be blank

Nombres

Apellidos

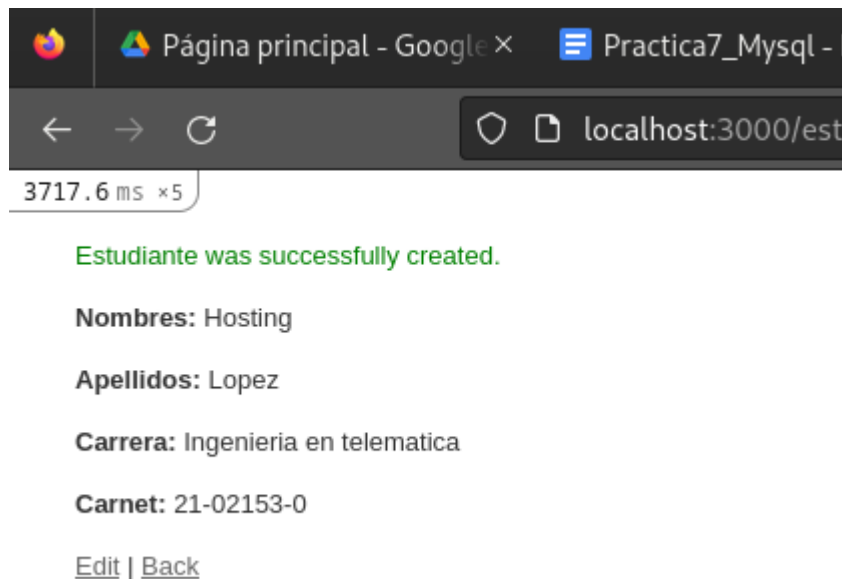
Carrera

Carnet

[Create Estudiante](#)

[Back](#)

18. Agregamos a un nuevo estudiante



19. Verificar que se haya creado correctamente el usuario en mysql, abrir el phpmyadmin y hacer un select a la tabla estudiante , de la base de datos en la que se está trabajando.

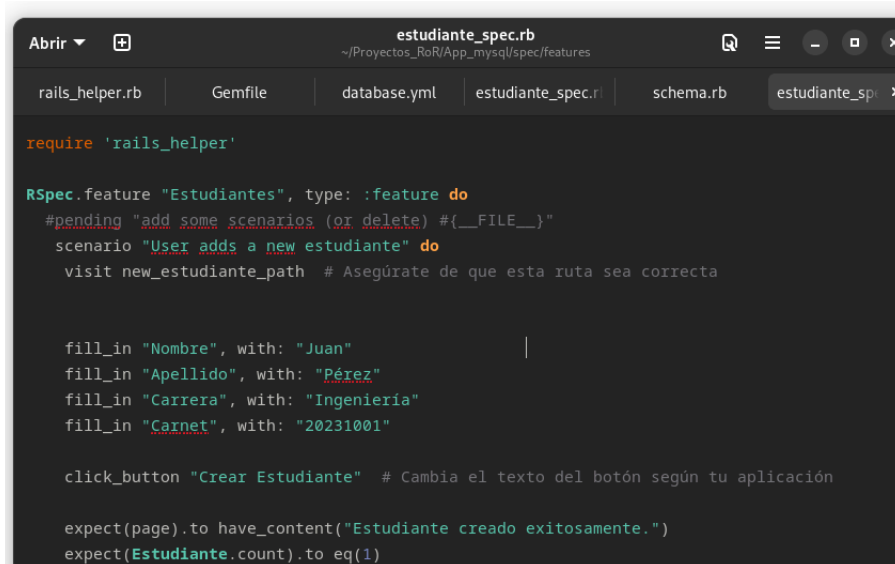
id	nombres	apellidos	carrera	carnet
1	Hosting	Lopez	Ingenieria en telematica	21-02153-0

20. Pruebas por comportamiento BDD.

20.1 Para hacer testing primero crear el directorio en donde se guardan los features.

```
Hosting@debian App_mysql $ rails g rspec:feature estudiante
Running via Spring preloader in process 21936
create spec/features/estudiante_spec.rb
```

21. Abrir el archivo spec/features/user_spec.rb y agregar el siguiente código, para hacer test al comportamiento al momento de agregar un nuevo estudiante.



```

Hosting@debian App_mysql $bundle exec rspec spec/features/estudiante_spec.rb

Estudiantes
  User adds a new estudiante

Finished in 0.65011 seconds (files took 3.67 seconds to load)
1 example, 0 failures

```

22. Otra prueba que se puede realizar en relación al comportamiento, es cuando el usuario presione click en el link back se debe mostrar la página principal de la aplicación donde se muestran todos los usuarios almacenados.

22.1 Copiar el siguiente código en el archivo spec/features/estudiante_spec.rb

```

scenario "User clicks back link and is redirected to main page" do
  visit new_estudiante_path
  click_link "Back"
  expect(current_path).to eq(estudiantes_path)
  expect(page).to have_content("Lista de Estudiantes")
end

```

```

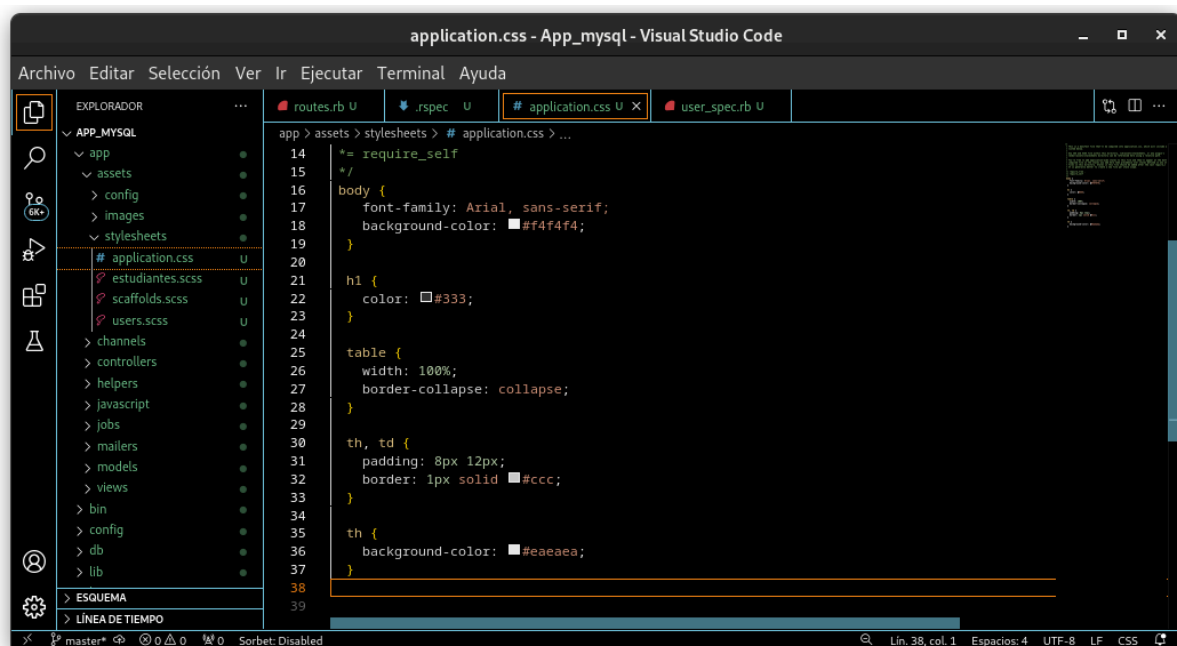
Hosting@debian App_mysql $bundle exec rspec spec/features/estudiante_spec.rb

Estudiantes
  User adds a new estudiante
  User clicks back link and is redirected to main page

Finished in 0.65613 seconds (files took 3.04 seconds to load)
2 examples, 0 failures

```

23. Agregamos css



134.4 ms

Estudiantes

Nombres	Apellidos	Carrera	Carnet			
Hosting	Lopez	Ingenieria en telematica	21-02153-0	Show	Edit	Destroy
Carlos Jose	Perez Garcia	Administracion	22-87788-0	Show	Edit	Destroy

[New Estudiante](#)